

**This Page is Inserted by IFW Indexing and Scanning  
Operations and is not part of the Official Record**

**BEST AVAILABLE IMAGES**

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ BLACK BORDERS
- ☐ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES
- ☐ FADED TEXT OR DRAWING
- ☐ BLURRED OR ILLEGIBLE TEXT OR DRAWING
- ☐ SKEWED/SLANTED IMAGES
- ☐ COLOR OR BLACK AND WHITE PHOTOGRAPHS
- ☐ GRAY SCALE DOCUMENTS
- ☐ LINES OR MARKS ON ORIGINAL DOCUMENT
- ☐ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY
- ☐ OTHER: \_\_\_\_\_

**IMAGES ARE BEST AVAILABLE COPY.**

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
www.uspto.gov

| APPLICATION NO. | FILING DATE | FIRST NAMED INVENTOR | ATTORNEY DOCKET NO. | CONFIRMATION NO. |
|-----------------|-------------|----------------------|---------------------|------------------|
| 10/055,483      | 01/22/2002  | Jared W. Stark IV    | 10559-559001        | 7484             |

20985 7590 09/30/2004

FISH & RICHARDSON, PC  
12390 EL CAMINO REAL  
SAN DIEGO, CA 92130-2081

EXAMINER

HUISMAN, DAVID J

| ART UNIT | PAPER NUMBER |
|----------|--------------|
|----------|--------------|

2183

DATE MAILED: 09/30/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

|                              |                  |              |  |
|------------------------------|------------------|--------------|--|
| <b>Office Action Summary</b> | Application No.  | Applicant(s) |  |
|                              | 10/055,483       | STARK ET AL. |  |
|                              | Examiner         | Art Unit     |  |
|                              | David J. Huisman | 2183         |  |

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

### Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

### Status

- 1) ☒ Responsive to communication(s) filed on 22 January 2002.
- 2a) ☐ This action is FINAL.                      2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

### Disposition of Claims

- 4) ☒ Claim(s) 1-46 is/are pending in the application.
- 4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.
- 5) ☐ Claim(s) \_\_\_\_\_ is/are allowed.
- 6) ☒ Claim(s) 1-46 is/are rejected.
- 7) ☐ Claim(s) \_\_\_\_\_ is/are objected to.
- 8) ☐ Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

### Application Papers

- 9) ☒ The specification is objected to by the Examiner.
- 10) ☒ The drawing(s) filed on 22 January 2002 is/are: a) ☐ accepted or b) ☒ objected to by the Examiner.  
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).  
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

### Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All    b) ☐ Some \*    c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
2. ☐ Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.
3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

\* See the attached detailed Office action for a list of the certified copies not received.

### Attachment(s)

- |  |   |
|--|---|
| 1) <input checked="" type="checkbox"/> Notice of References Cited (PTO-892)                        | 4) <input type="checkbox"/> Interview Summary (PTO-413)                     |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948)               | Paper No(s)/Mail Date. _____  |
| 3) <input checked="" type="checkbox"/> Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) | 5) <input type="checkbox"/> Notice of Informal Patent Application (PTO-152) |
| Paper No(s)/Mail Date <u>22 October 2003</u> .   | 6) <input type="checkbox"/> Other: _____                                    |

### **DETAILED ACTION**

1. Claims 1-46 have been examined.

#### ***Papers Submitted***

2. It is hereby acknowledged that the following papers have been received and placed of record in the file: IDS as received on 10/22/2003.

#### ***Specification***

3. The title of the invention is not descriptive. A new title is required that is clearly indicative of the invention to which the claims are directed.
4. The abstract of the disclosure is objected to because of the following: In line 4, remove the word "that". In addition, the sentence beginning with "At least..." is a sentence fragment, and it should therefore be reworded. Finally, remove "10128865.doc" from the bottom of the abstract. The following guidelines illustrate the preferred layout for the specification of a utility application. These guidelines are suggested for the applicant's use.
5. As provided in 37 CFR 1.77(b), the specification of a utility application should include the following sections in order. Each of the lettered items should appear in upper case, without underlining or bold type, as a section heading. If no text follows the section heading, the phrase "Not Applicable" should follow the section heading:

(a) TITLE OF THE INVENTION.

(b) CROSS-REFERENCE TO RELATED APPLICATIONS.

(c) STATEMENT REGARDING FEDERALLY SPONSORED RESEARCH OR  
DEVELOPMENT.

Art Unit: 2183

- (d) INCORPORATION-BY-REFERENCE OF MATERIAL SUBMITTED ON A COMPACT DISC (See 37 CFR 1.52(e)(5) and MPEP 608.05. Computer program listings (37 CFR 1.96(c)), "Sequence Listings" (37 CFR 1.821(c)), and tables having more than 50 pages of text are permitted to be submitted on compact discs.) or  
REFERENCE TO A "MICROFICHE APPENDIX" (See MPEP § 608.05(a). "Microfiche Appendices" were accepted by the Office until March 1, 2001.)
- (e) BACKGROUND OF THE INVENTION.
  - (1) Field of the Invention.
  - (2) Description of Related Art including information disclosed under 37 CFR 1.97 and 1.98.
- (f) **BRIEF SUMMARY OF THE INVENTION.**
- (g) BRIEF DESCRIPTION OF THE SEVERAL VIEWS OF THE DRAWING(S).
- (h) DETAILED DESCRIPTION OF THE INVENTION.
- (i) CLAIM OR CLAIMS (commencing on a separate sheet).
- (j) ABSTRACT OF THE DISCLOSURE (commencing on a separate sheet).
- (k) SEQUENCE LISTING (See MPEP § 2424 and 37 CFR 1.821-1.825. A "Sequence Listing" is required on paper if the application discloses a nucleotide or amino acid sequence as defined in 37 CFR 1.821(a) and if the required "Sequence Listing" is not submitted as an electronic document on compact disc).

Correction is required, as the examiner has been unable to locate a "brief summary" section. See MPEP § 608.01(b).

6. The lengthy specification has not been checked to the extent necessary to determine the presence of all possible minor errors. Applicant's cooperation is requested in correcting any errors of which applicant may become aware in the specification.

7. The disclosure is objected to because of the following informalities: Replace all occurrences of "dependences" with --dependencies--. On page 4, paragraph 17, remove the comma after the period. On page 7, paragraph 23, at the beginning of the third sentence, replace "Each" with --As shown in Figure 4, each--. On page 8, paragraph 24, replace the period after "SHIFT" with a comma. On page 9, paragraph 26, replace "re quest" with --request--. On page 10, paragraph 27, replace "N- 1" with --N-1--. On page 12, please reword the second sentence of paragraph 32, as it appears to be a fragment. On page 16, replace "select-I" with --select-1--.

Art Unit: 2183

Appropriate correction is required.

### *Drawings*

8. The drawings are objected to because of the following minor informalities: In Fig.4, the examiner requests that the arrows corresponding to number 54 be more neatly illustrated (not overlapping). And, in Fig.5, the examiner requests that applicant label the group of seven fields that provide data to the OR gates. Corrected drawing sheets in compliance with 37 CFR 1.121(d) are required in reply to the Office action to avoid abandonment of the application. Any amended replacement drawing sheet should include all of the figures appearing on the immediate prior version of the sheet, even if only one figure is being amended. The figure or figure number of an amended drawing should not be labeled as "amended." If a drawing figure is to be canceled, the appropriate figure must be removed from the replacement sheet, and where necessary, the remaining figures must be renumbered and appropriate changes made to the brief description of the several views of the drawings for consistency. Additional replacement sheets may be necessary to show the renumbering of the remaining figures. The replacement sheet(s) should be labeled "Replacement Sheet" in the page header (as per 37 CFR 1.84(c)) so as not to obstruct any portion of the drawing figures. If the changes are not accepted by the examiner, the applicant will be notified and informed of any required corrective action in the next Office action. The objection to the drawings will not be held in abeyance.

*Claim Objections*

9. Claim 10 is objected to because of the following informalities: In line 2, replace “includes;” with --includes:--. In the last line of claim 10, replace “forexecution” with --for execution--. Appropriate correction is required.
10. Claim 11 is objected to because of the following informalities: In line 2, replace “includes;” with --includes:--. Appropriate correction is required.
11. Claim 13 is objected to because of the following informalities: In line 3, insert --is-- after “instructions”. Appropriate correction is required.
12. Claim 15 is objected to because of the following informalities: It is not clear to the examiner how scheduling logic has an execution latency, as scheduling logic does not execute instructions. The examiner recommends rewording this in a more clear fashion. For instance, in lines 3 and 6, replace “execution” with --scheduling--. Appropriate correction is required.
13. Claim 16 is objected to because of the following informalities: It is not clear to the examiner how a dynamic instruction scheduler executes instructions. Instead, a functional unit executes instructions. The examiner recommends rewording this in a more clear fashion. For instance, replace “dynamic instruction scheduler” with --functional unit--. And, in line 4, replace “execution” with --scheduling--. Appropriate correction is required.
14. Claim 17 is objected to because of the following informalities: In lines 2 and 4, replace “includes;” with --includes:-- and “including;” with --including:--, respectively.. Appropriate correction is required.
15. Claim 23 is objected to because of the following informalities: In line 4, replace “including;” with --including:--. Appropriate correction is required.

***Claim Rejections - 35 USC § 102***

16. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

17. Claims 1-4, 7, 10, 12-20, 23-26, 29, 32-34, 38-42, and 46 are rejected under 35

U.S.C. 102(b) as being anticipated by Stark et al., “On Pipelining Dynamic Instruction Scheduling Logic,” (as disclosed by applicant and herein referred to as Stark).

18. Referring to claim 1, Stark has taught a processor comprising:

a) a wakeup loop to hold scheduler instructions including unexecuted instructions, and to indicate ready instructions of the unexecuted instructions that may be ready to be executed. See page 59, column 1, paragraphs 1-2. Note that when an instruction is ready to be executed, it is woken up and eventually selected.

b) at least one of the unexecuted instructions to wakeup and notify at least another of the unexecuted instructions to speculatively wakeup. See Fig.7 and the first full paragraph in column 2 on page 61, for instance, and note that a first instruction will wakeup a dependent instruction (in this case, XOR wakes up SUB and SUB wakes up ADD).

c) a select loop to select at least one of the ready instructions for execution. See the first sentence of section 3.3.

19. Referring to claim 2, Stark has taught a processor as described in claim 1. Stark has further taught a collision handling technique. See the second sentence of section 3.3 and note



Art Unit: 2183

that if multiple instructions are ready for execution (collision), some scheme is employed to handle the collision and select one of the instructions.

20. Referring to claim 3, Stark has taught a processor as described in claim 2. Stark has further taught that the collision handling technique includes a predict another wakeup technique. See section 4.3 and note that if multiple instructions become ready for execution at the same time, then the older one will be selected. This is consistent with applicant's disclosed PAW technique in that older instructions will execute first.

21. Referring to claim 4, Stark has taught a processor as described in claim 3. Stark has further taught that the predict another wakeup technique includes a PAW vector. See the end of section 4.3 and note that an instruction age (vector) is associated with each instruction. The age is then used to determine which instruction is the oldest so that it is next to execute.

22. Referring to claim 7, Stark has taught a processor as described in claim 1. Stark has further taught that the scheduler instructions include executed instructions. See section 4.5.2, and note that instructions may be re-executed. That is, the instructions in the reservation stations that are to be re-executed are actually executed instructions in that they have been executed previously.

23. Referring to claim 10, Stark has taught a processor as described in claim 1. Stark has further taught that the wakeup loop includes:

a) a wakeup array to hold the scheduler instructions. See Fig.3 and note that waiting instructions are stored in reservation stations (arrays). And the wakeup logic is part of the reservation stations. See the first paragraph on page 59.

Art Unit: 2183

b) wakeup logic to indicate the at least one of the unexecuted instructions that may be ready to be selected for execution. See Fig.6 and its description on page 60, and note that when an instruction's sources are ready, the instruction will be woken up and marked as ready.

24. Referring to claim 12, Stark has taught a processor as described in claim 1. Stark has further taught that the wakeup logic is selected from the group consisting of AND/OR array logic, CAM-style logic, and RAM-style logic. See Fig.10 and note the use of AND/OR gates.

25. Referring to claim 13, Stark has taught a processor as described in claim 1. Stark has further taught that the select loop includes select logic to generate a grant vector indicating at least one of the unexecuted instructions is speculatively ready and granted execution. See Fig.10 and section 4.2 and the first paragraph of section 4.3, and note that when an instruction is speculatively ready and granted execution, a destination tag (vector) is generated and transmitted on the destination tag bus.

26. Referring to claim 14, Stark has taught a processor as described in claim 1. Stark has further taught that the wakeup loop is pipelined over at least two cycles. See sections 3.5. and 4.

27. Referring to claim 15, Stark has taught a processor comprising a select-free scheduler including wakeup and select logic having a total execution latency, to schedule instructions for functional units that execute instructions having an execution latency that is less than the total execution latency of the wakeup and select logic. See section 3.5 and note that the wakeup and select logic (scheduling logic) requires two cycles because it is pipelined. Furthermore, from Fig.8, it can be seen that the ADD and OR instructions require one execution cycle. Therefore, the execution latency of the functional units is less than the total latency required for scheduling.

Art Unit: 2183

28. Referring to claim 16, Stark has taught a processor as described in claim 15. Stark has further taught a dynamic instruction scheduler to execute instructions that have an execution latency that is at least equal to the total execution latency of the wakeup and select logic. Recall from section 3.5 that scheduling requires two cycles. And, Fig.8 shows that a functional unit takes three cycles to execute a load instruction. Therefore, the functional unit latency is at least equal to the latency of the scheduling logic.

29. Referring to claim 17, Stark has taught a processor as described in claim 15. Stark has further taught that the select-free scheduler includes: a wakeup stage to hold scheduler instructions including unexecuted instructions (see the first paragraph on page 59), the wakeup stage including:

- a) a wakeup array having resource vectors corresponding to the unexecuted instructions to indicate dependencies upon resources. See Fig.10 and note the resource vector which includes information regarding when an instruction's parents and/or grandparents are ready.
- b) wakeup logic to indicate at least one of the unexecuted instructions that may be ready to be selected for execution. See Fig.10 and the first full paragraph in column 2 on page 63. Note that when an instruction is speculatively ready a request is made for execution.
- c) select logic including speculative wakeup to indicate at least one of the unexecuted instructions that may be ready to be selected for execution. See Fig.10 and note the select logic, and also see the first few sentences of section 4.3. The logic indicates that the instruction may be ready to execute but a confirm signal is also required which indicates it is ready to execute.

30. Referring to claim 18, Stark has taught a processor as described in claim 17. Stark has taught that the processor further comprises a collision handling technique. See the second

Art Unit: 2183

sentence of section 3.3 and note that if multiple instructions are ready for execution (collision), some scheme is employed to handle the collision and select one of the instructions.

31. Referring to claim 19, Stark has taught a processor as described in claim 18. Stark has further taught that the collision handling technique includes a predict another wakeup technique. See section 4.3 and note that if multiple instructions become ready for execution at the same time, then the older one will be selected. This is consistent with applicant's disclosed PAW technique in that older instructions will execute first.

32. Referring to claim 20, Stark has taught a processor as described in claim 19. Stark has further taught that the predict another wakeup technique includes a PAW vector. See the end of section 4.3 and note that an instruction age (vector) is associated with each instruction. The age is then used to determine which instruction is the oldest so that it is next to execute.

33. Referring to claim 23, Stark has taught a system comprising a processor including:

a) a random access memory (RAM) device. It is inherent that a RAM device exists within Stark. This is clear because Stark discloses load instructions (see Fig.8). Loads inherently access RAM if they miss a cache, and Stark discloses cache misses. See the first paragraph of column 2 on page 58.

b) a processor in communication with the RAM device, the processor including:

b1) a wakeup loop to hold scheduler instructions including unexecuted instructions, and to indicate ready instructions of the unexecuted instructions that may be ready to be executed. See Fig.10, section 4.2, and the first paragraph on page 59.

b2) at least one of the unexecuted instructions to wakeup and notify at least another of the unexecuted instructions to speculatively wakeup. See Fig.10 and the first full paragraph

Art Unit: 2183

in column 2 on page 63, note that an unexecuted instruction may wake up and request execution when its parents are ready or all of its grandparents are ready.

b3) a select loop to select at least one of the ready instructions for execution. See section 4.3 and note that an instruction is selected for execution when its parents are ready.

34. Referring to claim 24, Stark has taught a system as described in claim 23. Stark has taught that the system further comprises a collision handling technique. See the second sentence of section 3.3 and note that if multiple instructions are ready for execution (collision), some scheme is employed to handle the collision and select one of the instructions.

35. Referring to claim 25, Stark has taught a processor as described in claim 24. Stark has further taught that the collision handling technique includes a predict another wakeup technique. See section 4.3 and note that if multiple instructions become ready for execution at the same time, then the older one will be selected. This is consistent with applicant's disclosed PAW technique in that older instructions will execute first.

36. Referring to claim 26, Stark has taught a processor as described in claim 25. Stark has further taught that the predict another wakeup technique includes a PAW vector. See the end of section 4.3 and note that an instruction age (vector) is associated with each instruction. The age is then used to determine which instruction is the oldest so that it is next to execute.

37. Referring to claim 29, Stark has taught a system as described in claim 23. Stark has further taught that the scheduler instructions include executed instructions. See section 4.5.2, and note that instructions may be re-executed. That is, the instructions in the reservation stations that are to be re-executed are actually executed instructions in that they have been executed previously.

Art Unit: 2183

38. Referring to claim 32, Stark has taught a method of issuing requesting instructions to an execution unit, comprising:

- a) speculatively setting an indicator to indicate a requesting instruction is ready to be selected for execution. See Fig.10 and sections 4.2 and 4.3. Note that if an instruction's parents are ready, for instance, then the instruction wakes up speculatively, by speculatively setting a request signal. The confirm line is later asserted, signifying the instruction is truly ready for execution.
- b) during a cycle, selecting a predetermined number of the requesting instructions having a set indicator. See the first paragraph of section 4.3 and section 3.3. Note that one ready instruction is selected for execution based on some heuristic. For instance, as section 4.3 discloses, request priority is one factor that may be taken into consideration when choosing a ready instruction for execution.
- c) resetting the indicator of the requesting instructions that are selected. See the last sentence in section 3.3, and note that this is inherent. If an indicator is used to say the instruction is ready, then it must be cleared when the instruction is finally selected for execution. Otherwise, the executed instruction will continue to appear to be ready, and possibly be selected over and over again for redundant execution.

39. Referring to claim 33, Stark has taught a method as described in claim 32. Stark has further taught that the predetermined number of selected instructions is one. See section 3.3 and 4.3.

40. Referring to claim 34, Stark has taught a method as described in claim 32. Stark has taught that the method further comprises handling collisions. See the second sentence of section

Art Unit: 2183

3.3 and note that if multiple instructions are ready for execution (collision), some scheme is employed to handle the collision and select one of the instructions.

41. Referring to claim 38, Stark has taught a method as described in claim 34. Stark has taught delaying setting the indicator based on predicting wakeup of another instruction. See section 4.3 and note that if multiple instructions are predicted to wakeup, then the older one will be selected. Therefore, the indication will not be set for the younger instruction so that it will not be selected for execution before the older instruction.

42. Referring to claim 39, Stark has taught a method of issuing unexecuted instructions to an execution unit, comprising:

a) generating resource vectors corresponding to the unexecuted instructions, the resource vectors including resource indicators to indicate availability of resources. See Fig. 10 and note the resource vector which includes information regarding when an instruction's parents and/or grandparents are ready. In essence, if the parents or grandparents are ready, it means that the resources on which the instruction is dependent on will be available soon also (as they are produced by the grandparents and parents), and therefore, a resource indicator (request line) is used to make such an indication.

b) speculatively setting the resource indicators to indicate resources associated with corresponding ones of the unexecuted instructions are available so that the corresponding ones of the unexecuted instructions are ready to be executed. See the second paragraph in section 4.

Also, from Fig. 10, note that the request line is set which indicates that the resources are likely to be available soon and therefore, the instruction should wake up.

Art Unit: 2183

c) selecting a predetermined number of the corresponding ones of the unexecuted instructions.

See the first paragraph of section 4.3 and section 3.3. Note that one ready instruction is selected for execution based on some heuristic. For instance, as section 4.3 discloses, request priority is one factor that may be taken into consideration when choosing a ready instruction for execution.

43. Referring to claim 40, Stark has taught a method as described in claim 39. Stark has further taught resetting the resource indicators corresponding to the unexecuted instructions that are selected. See the last sentence in section 3.3, and note that this is inherent. If an indicator is used to say the instruction is ready, then it must be cleared when the instruction is finally selected for execution. Otherwise, the executed instruction will continue to appear to be ready, and possibly be selected over and over again for redundant execution.

44. Referring to claim 41, Stark has taught a method as described in claim 39. Stark has further taught that the predetermined number of selected instructions is one. See section 3.3 and 4.3.

45. Referring to claim 42, Stark has taught a method as described in claim 39. Stark has further taught handling collisions. See the second sentence of section 3.3 and note that if multiple instructions are ready for execution (collision), some scheme is employed to handle the collision and select one of the instructions.

46. Referring to claim 46, Stark has taught a method as described in claim 42. Stark has taught delaying setting the resource indicators based on predicting wakeup of another instruction. See section 4.3 and note that if multiple instructions are predicted to wakeup, then the older one will be selected. Therefore, the indicators (request and confirm lines) will not be set for the younger instruction so that it will not be selected for execution before the older instruction.



***Claim Rejections - 35 USC § 103***

47. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

48. Claims 5-6, 8-9, 11, 21-22, 27-28, 30-31, 35-37, and 43-45 are rejected under 35 U.S.C. 103(a) as being unpatentable over Stark, as applied above, in view of Hennessy and Patterson, "Computer Architecture - A Quantitative Approach, 2<sup>nd</sup> Edition," 1996 (herein referred to as Hennessy).

49. Referring to claim 5, Stark has taught a processor as described in claim 2. Stark has not taught that the collision handling technique includes a scoreboard to indicate whether dependent instructions of an executed instruction have executed. However, Hennessy has taught such a concept. See page 248 and note that the scoreboard indicates that the MULTD, SUBD, and ADDD instructions have all finished completion, and these instructions are dependent instructions of instructions that have already executed. A scoreboard allows instructions to execute out-of-order which allows instructions to execute as soon as their operands are available. Consequently, less stalling is required. As a result it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Stark to include a scoreboard.

50. Referring to claim 6, Stark in view of Hennessy has taught a processor as described in claim 5. Hennessy has further taught a pileup victims vector is computed based on the scoreboard. See page 247 and note from the "functional unit status" table that some instructions

Art Unit: 2183

are not ready to be executed and so they are piling up in the scoreboard. These instructions include all of the instructions with  $R_j$  or  $R_k = \text{No}$ . This means that their operands are not ready and therefore the instructions are just waiting. They are detected by the values of  $R_j$  and  $R_k$ , which would likely be a value of 0 (for No) and 1 (for Yes).

51. Referring to claim 8, Stark has taught a processor as described in claim 1. Stark has not taught that the select loop generates a collision victim vector to identify collision victims. However, Hennessy has taught the concept of generating a vector within a scoreboard which is used to determine a collision has occurred. See page 247 and note that the SUBD and ADDD instructions collide because they both require the same functional unit (see caption). Since the SUBD is older and also the parent of the ADDD, it will be issued first, thereby generating the vector associated with the SUBD instruction. The vector comprises bits which represent the fields shown in the "functional unit status" and "instruction status" table. In this situation, the important data is that which specifies the ADD unit (which is used to execute the SUBD and ADDD instructions) is busy. By saying it is busy, the ADDD cannot wakeup and be issued for execution (as shown in the "instruction status" table). The bits used to specify the ADDD instruction has not issued indicates that it has collided with another instruction. A scoreboard allows instructions to execute out-of-order which allows instructions to execute as soon as their operands are available. See page 242. Consequently, less stalling is required. As a result it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Stark to include a scoreboard which is used to detect collision victims.

52. Referring to claim 9, Stark in view of Hennessy has taught a processor as described in claim 8. Hennessy has further taught that the collision victim vector is communicated to the

Art Unit: 2183

wakeup loop. Again, see page 247. More specifically, this vector is used to direct wakeup of instructions. For instance, when the SUBD instruction is done with the ADD unit, then it will no longer be busy and therefore, the ADDD instruction may be woken up and issued.

53. Referring to claim 11, Stark has taught a processor as described in claim 10.

a) Stark has further taught that the wakeup array includes a resource vector corresponding to each of the scheduler instructions to indicate dependencies upon resources. See Fig. 10 and note the resource vector which includes information regarding when an instruction's parents and/or grandparents are ready.

b) Stark has not taught a PAW vector to indicate the resources needed by earlier instructions in the wakeup array. However, Hennessy has taught the concept of bits used to indicate resources needed by earlier instructions. See page 247 and note the "functional unit status" table. Note that bits are used to indicate what functional units are busy and what resources are required by instructions. This type of information allows instructions to execute out-of-order which allows instructions to execute as soon as their operands are available. See page 242. Consequently, less stalling is required. As a result it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Stark to provide a PAW vector for monitoring resources of earlier instructions.

54. Referring to claim 21, Stark has taught a processor as described in claim 18. Stark has not taught that the collision handling technique includes a scoreboard to indicate whether dependent instructions of an executed instruction have executed. However, Hennessy has taught such a concept. See page 248 and note that the scoreboard indicates that the MULTD, SUBD, and ADDD instructions have all finished completion, and these instructions are dependent

Art Unit: 2183

instructions of instructions that have already executed. A scoreboard allows instructions to execute out-of-order which allows instructions to execute as soon as their operands are available. Consequently, less stalling is required. As a result it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Stark to include a scoreboard.

55. Referring to claim 22, Stark in view of Hennessy has taught a processor as described in claim 21. Hennessy has further taught a pileup victims vector is computed based on the scoreboard. See page 247 and note from the "functional unit status" table that some instructions are not ready to be executed and so they are piling up in the scoreboard. These instructions include all of the instructions with Rj or Rk = No. This means that their operands are not ready and therefore the instructions are just waiting. They are detected by the values of Rj and Rk, which would likely be a value of 0 (for No) and 1 (for Yes).

56. Referring to claim 27, Stark has taught a processor as described in claim 24. Stark has not taught that the collision handling technique includes a scoreboard to indicate whether dependent instructions of an executed instruction have executed. However, Hennessy has taught such a concept. See page 248 and note that the scoreboard indicates that the MULTD, SUBD, and ADDD instructions have all finished completion, and these instructions are dependent instructions of instructions that have already executed. A scoreboard allows instructions to execute out-of-order which allows instructions to execute as soon as their operands are available. Consequently, less stalling is required. As a result it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Stark to include a scoreboard.

57. Referring to claim 28, Stark in view of Hennessy has taught a processor as described in claim 27. Hennessy has further taught a pileup victims vector is computed based on the

Art Unit: 2183

scoreboard. See page 247 and note from the “functional unit status” table that some instructions are not ready to be executed and so they are piling up in the scoreboard. These instructions include all of the instructions with  $R_j$  or  $R_k = \text{No}$ . This means that their operands are not ready and therefore the instructions are just waiting. They are detected by the values of  $R_j$  and  $R_k$ , which would likely be a value of 0 (for No) and 1 (for Yes).

58. Referring to claim 30, Stark has taught a processor as described in claim 23. Stark has not taught that the select loop generates a collision victim vector to identify collision victims. However, Hennessy has taught the concept of generating a vector within a scoreboard which is used to determine a collision has occurred. See page 247 and note that the SUBD and ADDD instructions collide because they both require the same functional unit (see caption). Since the SUBD is older and also the parent of the ADDD, it will be issued first, thereby generating the vector associated with the SUBD instruction. The vector comprises bits which represent the fields shown in the “functional unit status” and “instruction status” table. In this situation, the important data is that which specifies the ADD unit (which is used to execute the SUBD and ADDD instructions ) is busy. By saying it is busy, the ADDD cannot wakeup and be issued for execution (as shown in the “instruction status” table). The bits used to specify the ADDD instruction has not issued indicates that it has collided with another instruction. A scoreboard allows instructions to execute out-of-order which allows instructions to execute as soon as their operands are available. See page 242. Consequently, less stalling is required. As a result it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Stark to include a scoreboard which is used to detect collision victims.

Art Unit: 2183

59. Referring to claim 31, Stark in view of Hennessy has taught a processor as described in claim 30. Hennessy has further taught that the collision victim vector is communicated to the wakeup loop. Again, see page 247. More specifically, this vector is used to direct wakeup of instructions. For instance, when the SUBD instruction is done with the ADD unit, then it will no longer be busy and therefore, the ADDD instruction may be woken up and issued.

60. Referring to claim 35, Stark has taught a method as described in claim 34. Stark has not taught that handling collisions includes generating a collision victims vector. However, Hennessy has taught the concept of generating a vector within a scoreboard which is used to determine a collision has occurred. See page 247 and note that the SUBD and ADDD instructions collide because they both require the same functional unit (see caption). Since the SUBD is older and also the parent of the ADDD, it will be issued first, thereby generating the vector associated with the SUBD instruction. The vector comprises bits which represent the fields shown in the "functional unit status" and "instruction status" table. In this situation, the important data is that which specifies the ADD unit (which is used to execute the SUBD and ADDD instructions ) is busy. By saying it is busy, the ADDD cannot wakeup and be issued for execution (as shown in the "instruction status" table). The bits used to specify the ADDD instruction has not issued indicates that it has collided with another instruction. A scoreboard allows instructions to execute out-of-order which allows instructions to execute as soon as their operands are available. See page 242. Consequently, less stalling is required. As a result it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Stark to include a scoreboard which is used to detect collision victims.

Art Unit: 2183

61. Referring to claim 36, Stark has taught a method as described in claim 34. Stark has not taught that handling collisions includes generating a pileup victims vector. However, Hennessy has taught the concept of generating a vector within a scoreboard which is used to determine a pileup. See page 247 and note that the SUBD and ADDD instructions collide because they both require the same functional unit (see caption). Since the SUBD is older and also the parent of the ADDD, it will be issued first, thereby generating the vector associated with the SUBD instruction. The ADDD, on the other hand, must wait until the SUBD instruction finishes with the execution unit, and therefore it is a pileup victim (i.e., there is a pile of instructions waiting to use the ADD unit). The pileup vector comprises bits which represent the fields shown in the "instruction status" table. In this situation, the important data is that which specifies that the ADDD instruction has not issued. A scoreboard allows instructions to execute out-of-order which allows instructions to execute as soon as their operands are available. See page 242. Consequently, less stalling is required. As a result it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Stark to include a scoreboard which is used to detect collision victims.

62. Referring to claim 37, Stark in view of Hennessy has taught a method as described in claim 36. Hennessy has further taught that generating a pileup victims vector includes reading a scoreboard and identifying pileup victims based upon the scoreboard. See page 247.

63. Referring to claim 43, Stark has taught a method as described in claim 42. Stark has not taught that handling collisions includes generating a collision victims vector. However, Hennessy has taught the concept of generating a vector within a scoreboard which is used to determine a collision has occurred. See page 247 and note that the SUBD and ADDD

Art Unit: 2183

instructions collide because they both require the same functional unit (see caption). Since the SUBD is older and also the parent of the ADDD, it will be issued first, thereby generating the vector associated with the SUBD instruction. The vector comprises bits which represent the fields shown in the “functional unit status” and “instruction status” table. In this situation, the important data is that which specifies the ADD unit (which is used to execute the SUBD and ADDD instructions ) is busy. By saying it is busy, the ADDD cannot wakeup and be issued for execution (as shown in the “instruction status” table). The bits used to specify the ADDD instruction has not issued indicates that it has collided with another instruction. A scoreboard allows instructions to execute out-of-order which allows instructions to execute as soon as their operands are available. See page 242. Consequently, less stalling is required. As a result it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Stark to include a scoreboard which is used to detect collision victims.

64. Referring to claim 44, Stark has taught a method as described in claim 42. Stark has not taught that handling collisions includes generating a pileup victims vector. However, Hennessy has taught the concept of generating a vector within a scoreboard which is used to determine a pileup. See page 247 and note that the SUBD and ADDD instructions collide because they both require the same functional unit (see caption). Since the SUBD is older and also the parent of the ADDD, it will be issued first, thereby generating the vector associated with the SUBD instruction. The ADDD, on the other hand, must wait until the SUBD instruction finishes with the execution unit, and therefore it is a pileup victim (i.e., there is a pile of instructions waiting to use the ADD unit). The pileup vector comprises bits which represent the fields shown in the “instruction status” table. In this situation, the important data is that which specifies that the



Art Unit: 2183

ADDD instruction has not issued. A scoreboard allows instructions to execute out-of-order which allows instructions to execute as soon as their operands are available. See page 242.

Consequently, less stalling is required. As a result it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Stark to include a scoreboard which is used to detect collision victims.

65. Referring to claim 45, Stark in view of Hennessy has taught a method as described in claim 44. Hennessy has further taught that generating a pileup victims vector includes reading a scoreboard and identifying pileup victims based upon the scoreboard. See page 247.

### *Conclusion*

Any inquiry concerning this communication or earlier communications from the examiner should be directed to David J. Huisman whose telephone number is (703) 305-7811. The examiner can normally be reached on Monday-Friday (8:00-4:30).

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Eddie Chan can be reached on (703) 305-9712. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

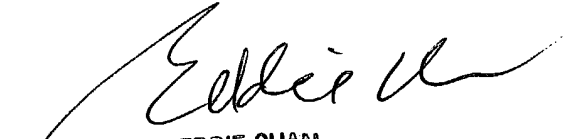
Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Art Unit: 2183

DJH

David J. Huisman

September 2, 2004



EDDIE CHAN  
SUPERVISORY PATENT EXAMINER  
TECHNOLOGY CENTER 2100